ORACLE

Configuring Oracle Database Clients for OID and OUD Directory Naming

Centralizing network names and addresses in Oracle Internet Directory (OID) and Oracle Unified Directory (OUD)

August, 2022
Copyright © 2022, Oracle and/or its affiliates

Table of contents

1. Introduction

When applications connect to Oracle Database, they need to use a connect descriptor that contains information such as the host and service name of the database. The connect descriptor can be specified by the application in various ways. For example, it can be hard coded in the application connection request or the application can pass an identifier that is mapped to a connect descriptor stored in a tnsnames.ora configuration file. An alternative to using a tnsnames.ora file is for the connect descriptor to be looked up using an external mapping service. One of the available services is Directory Naming.

Directory Naming centralizes network names and addresses in a single place, facilitating easy administration of name changes and updates. This eliminates the need administrators to change connect descriptors stored in tnsnames.ora files. In large organizations there could be hundreds, or even thousands, of database applications and tnsnames.ora files.

Oracle has two directory server products, Oracle Internet Directory (OID) and Oracle Unified Directory (OUD). This technical brief describes how to configure name resolution after they have been installed. The products can be download from Identity & Access Management Downloads. To install OID 12.2.1.4, follow this installation guide. To install OUD 12.2.1.4.0, follow this installation guide.

2. Setting up clients for OID and OUD Oracle Directory Naming

Once installed, OID and OUD will have a schema used for Directory Naming objects. Use the Oracle Net Configuration Assistant tool ("NetCA") to create the client configuration file to use Directory Naming. This tool allows you to select a naming context which will contain an OracleContext object. All Directory Naming objects will be created under the OracleContext.

To start configuration:

1. Run $ORACLE_HOME/bin/netca
2. Select the "Directory Usage Configuration" option.

Copyright © 2022, Oracle and/or its affiliates

3. In the Directory Type dropdown, select the “Oracle Internet Directory” option for either the OID or OUD Directory servers.
4. Enter the OID or OUD Directory Server details.
5. Select a naming context.

On completion, NetCA creates a file ldap.ora. The file will be in the directory $TNS_ADMIN, if that environment variable is set. Otherwise it will be in $ORACLE_HOME/network/admin or in $ORACLE_BASE_HOME/network/admin (if the installation is a read-only ORACLE_HOME).

An example ldap.ora file is:

```
DIRECTORY_SERVERS = (oidserver.example.com:389:636)
DEFAULT_ADMIN_CONTEXT = "DC=example,DC=com"
DIRECTORY_SERVER_TYPE = OID
```

Copy the ldap.ora file to all the client machines that will use directory naming, or alternatively run NetCA again on all those machines.

For reference, the Oracle Net Configuration Assistant documentation is here.

3. Creating Entries in Directory Servers

To create and manage naming entries in your directory server you must first create one directory user. This user manages the naming entries stored in the OracleContext set up above. Follow the directory server documentation to create a user. For example, in OID 12.2.1.4 follow here. In OUD follow here. The examples below assume the name is "mynaminguser".

You can then use the Oracle Net Manager tool (“NetManager”), use command line tools, or a REST API to create and manage naming entries as shown in the next sections. Using the REST API is discussed in the documentation REST API for Oracle Unified Directory Data Management.

3.1 Creating naming entries with Oracle Net Manager

Follow these steps:

1. Run $ORACLE_HOME/bin/netmgr

   If the ldap.ora file created in section 2 is in the correct location then NetManager will show a Directory icon in the tree navigator. If you don’t see the icon, make sure that ldap.ora is in $ORACLE_HOME/network/admin or $ORACLE_BASE_HOME/network/admin, if either location exists, or alternatively make sure that the TNS_ADMIN environment variable is set to the directory containing ldap.ora.

   If there is still no Directory subtree, use the command line tools in section 3.2.

2. Click and expand the Directory subtree.

   You will be prompted for a directory server user’s credentials. Enter the credentials created at the start of section 3.

3. Select the Service Naming subtree.

Copyright © 2022, Oracle and/or its affiliates

4. Click the Green Plus icon on the left hand menu.
5. Follow the Net Service Name wizard prompts to create a net service name and descriptor in the directory server.

For more information, see the Oracle Net Manager documentation.

3.2 Creating naming entries using command line tools

An alternative to Oracle Net Manager is to use LDAP command line tools like ldapadd and ldapdelete.

For example to add an alias "sales", create a file sales.ldif as shown below:

```
dn: cn=sales,cn=oraclecontext,dc=example,dc=com
objectclass: top
objectclass: orclNetService
orclnetdescstring:
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=databasehost)(PORT=152
1))(CONNECT_DATA=(SERVICE_NAME=mydbservice.us.example.com)))
cn: sales
```

This example assumes OracleContext is under "dc=example,dc=com". Adjust orclnetdescstring to use your Oracle Database host, port and service name.

Then run ldapadd using the syntax:

```
ldapadd -h <ldap host> -p  <ssl port> -D <Bind DN>  -U 2 -q \
        -Q -W file:<wallet directory> -f <ldif file>
```

For example:

```
ldapadd -h mydirectoryserverhost -p 636 \
        -D "cn=mynaminguser,dc=example,dc=com" -U 2 -q \
        -Q -W file:/opt/oracle/wallets -f sales.ldif
```

If you are using TLS 1.2 two-way authentication, then the option -U 3 should be used instead.

To delete entries, use the ldapdelete command. The syntax is similar to ldapadd but you specify the alias to be deleted:

```
ldapdelete -h <ldap host> -p <ssl port> -D <Bind DN> -U 2 -q \
           -Q -W file:<wallet directory> <alias DN>
```

For example:

```
ldapdelete -h mydirectoryserverhost -p 636 \
           -D "cn=mynaminguser,dc=example,dc=com" -U 2 -q \
           -Q -W file:/opt/oracle/wallets sales
```

Again, -U 3 can be used if you are using TLS 1.2 two-way authentication.

4. Client-side configuration for Directory Naming

Continue configuring the client machines.

4.1 Enabling the naming method lookup

On all machines that will use directory naming, create or edit a sqlnet.ora configuration file. This file should be in the same directory as your ldap.ora file from section 2.

Copyright © 2022, Oracle and/or its affiliates

In the sqlnet.ora file, add LDAP to the NAMES.DIRECTORY_PATH parameter. Application connection descriptors are evaluated in order of the given naming methods in that parameter. For example, to try LDAP first, and then fallback to using Easy Connect syntax, and finally look up the connection string in a tnsnames.ora file:

```
NAMES.DIRECTORY_PATH = (LDAP, EZCONNECT, TNSNAMES)
```

If you do not have a NAMES.DIRECTORY_PATH entry, LDAP will still be used but will not be considered first.

4.2 Directory Server Authentication

By default, directory naming does anonymous binding. However if the directory server has disabled anonymous binding, then you should configure authentication to the directory server in one of the two following ways.

4.2.2 OID client certificate-based authentication

With OID, certificate based authentication can be used for directory server access. This method is not available for OUD.

Each sqlnet.ora file should enable authenticated binding and point to a wallet containing an X.509 client certificate. For example, add these lines to sqlnet.ora:

```
NAMES.LDAP_AUTHENTICATE_BIND = TRUE
WALLET_LOCATION =
          (SOURCE = (METHOD = FILE)
             (METHOD_DATA =
             (DIRECTORY =  <wallet directory>)
          )
```

The OID directory server needs to be configured to accept the certificate's DN as its username and to map to a directory entry.

4.2.3 Username and password-based authentication

Alternatively, instead of using certificate authentication, you can use a username and password to access an OID or OUD directory server. This authentication method is also available to non-Microsoft Windows applications using Active Directory. Applications must use Oracle Database 21c (or later) client libraries.

With this method, the directory server username and password are stored in a wallet. The client authenticates by passing these credentials over a one-way TLS connection to the directory server.

Edit your sqlnet.ora files to enable authenticated binding and set the bind method. For example, add these lines to the files:

Copyright © 2022, Oracle and/or its affiliates

```
NAMES.LDAP_AUTHENTICATE_BIND = TRUE
NAMES.LDAP_AUTHENTICATE_BIND_METHOD = SIMPLE
WALLET_LOCATION =
          (SOURCE = (METHOD = FILE)
             (METHOD_DATA =
             (DIRECTORY = <wallet directory>)
          )
```

Obtain the Directory Server certificate root CA certificate and store it in a wallet as shown below. The username and password will also need to be added.

1. Create a wallet, if you do not already have one:

   ```
   orapki wallet create \
       -wallet <directory to create the wallet in>
   ```

2. Add the certificate to the wallet:

   ```
   orapki wallet add -wallet <wallet directory> \
       -trusted_cert -cert <root CA certificate>
   ```

3. Add the username:

   ```
   mkstore -wrl <wallet directory> -createEntry \
       oracle.ldap.client.dn <DN of the user>
   ```

   For example:

   ```
   mkstore -wrl /opt/oracle/wallets -createEntry \
       oracle.ldap.client.dn "cn=user1,dc=acme,dc=com"
   ```

   With Active Directory, the DN user name can also be a User Principal Name or Down Level Logon Name (also known as an SAMAccountName).

   For example, when foo is the domain name and user1 is the user name:

   ```
   mkstore -wrl /opt/oracle/wallets -createEntry \
       oracle.ldap.client.dn "foo\user1"
   ```

   or

   ```
   mkstore -wrl /opt/oracle/wallets -createEntry \
       oracle.ldap.client.dn "user1@foo"
   ```

4. Add the password of the user:

   ```
   mkstore -wrl <wallet directory> -createEntry \
       oracle.ldap.client.password <password>
   ```

5. Make the wallet an auto-logon wallet:

   ```
   orapki wallet create -wallet <wallet directory> \
       -auto_login
   ```

5. Testing a connection

Configuration of the naming server is complete. You can verify connections by running SQL*Plus from one of your configured client machines. Make sure the network configuration files are in a default location, or that you have set the environment variable TNS_ADMIN to the directory that contains them. If you have a database user “scott” and have an alias “sales” in a sales.ldif file, you would connect like:

```
sqlplus scott@sales
```


Copyright © 2022, Oracle and/or its affiliates

6. Conclusion

This technical brief has shown how to configure Oracle Database clients for Directory Naming using OID or OUD.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Copyright © 2022, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

Disclaimer: If you are unsure whether your data sheet needs a disclaimer, read the revenue recognition policy. If you have further questions about your content and the disclaimer requirements, e-mail REVREC_US@oracle.com.

Copyright © 2022, Oracle and/or its affiliates